Universal Learning System for Embedded System Education and Promotion

Kai-Chao Yang, Yu-Tsang Chang, Chien-Ming Wu, Chun-Ming Huang, and Chin-Long Wey
National Chip Implementation Center
Hsinchu, Taiwan

Abstract—**In this article, the idea of the universal learning system for embedded systems is presented. The proposed system provides a complete learning environment consisting of the information collection center, preference estimation system, Q&A center, forum, and virtual classroom. The skeleton of the proposed system is a preference estimation system, which helps users know the relationship between different hardware kits and suggests suitable hardware kits to users to learn embedded systems. Then, the proposed system provides the virtual classroom and Q&A service for users to start their classes. Besides, users can share design samples and experience, and join discussions through the forum of the proposed system. For demonstration, three embedded hardware platforms are introduced and applied by the proposed learning system. The results show that most students feel the proposed learning system can effectively help with their embedded software design.**

Keywords—Embedded system; Distance learning; Computer science education

I. INTRODUCTION

Due to the rapid development of electronic technology and requirements of electronic markets, electronic products tend to become smaller, faster, and more popular. In Europe, nearly 50% of the 100 biggest companies have invested in embedded systems research [1]. This implies the increasing requirements of the talents of embedded systems. More and more educators put emphasize on this area [2]-[9] as well. However, the great diversity of electronic products and applications also lead to inconvenience in education of embedded systems because it is difficult to learn all software design skills from the same embedded hardware platform.

Therefore, more and more embedded hardware platforms are developed and used in different courses for various applications, such as data sensing or video compression. When learning embedded software design, students usually encounter three difficulties. First, software resources are dispersive or unavailable on the Internet, so it is difficult for students to search for these resources. Second, the selection of embedded hardware platforms might not only confuse students but also educators. Third, time and locations for learning are confined. This might cause great inconvenience to students and thus restrict the growth of manpower.

Considering the above problems, the idea of the universal learning system is proposed in this article. The proposed system is constructed and maintained by National Chip Implementation Center (CIC) [10] in Taiwan, a national research and service center supported by Taiwan government for IC education and promotion. The universal learning system contains five parts, the information collection center, preference estimation system, Q&A center, forum, and virtual classroom. The skeleton of the universal learning system is the preference estimation system, which helps users make comparisons and selections among different hardware kits. In the preference estimation system, the features of hardware kits are visualized and personalized according to the user preference. These features are mapped to points in a low-dimensional coordinate system, so a user can easily tell the relationship between the user preference and hardware kits. Finally the proposed system can suggest the most suitable hardware kit for the user. After a user selects the hardware kit to learn, he can take courses in the virtual classroom and search for relevant resources such as documents and design samples in the information collection center. Besides, the user can join discussions and share experience or resources in the forum. Moreover, the Q&A center in the proposed system can be viewed as a unified window between users and vendors. The experts in CIC help users answer questions or forward questions to the vendor. Therefore, the preference estimation system can provide a solution to let users learn embedded systems anytime and anywhere.

A case study is presented to demonstrate the performance of the universal learning system. Three embedded hardware platforms [11]-[13] are imported into the proposed system. For each hardware platform, a short course is carefully designed according to the features of this hardware platform. A user can choose the hardware platform and take the course according to the suggestion of the preference estimation system. The student feedback shows that most students agreed the proposed system could help their study in embedded software design.

This article is organized as follows. In the next section, we give an overview of the universal learning system. Then the preference estimation system is presented in Section 3. In Section 4, a case study is described. In the case study, three embedded hardware platforms are introduced and imported to the proposed system. Some discussions and student feedback are also given in the same section. In the last section, a summarization and future works are presented.

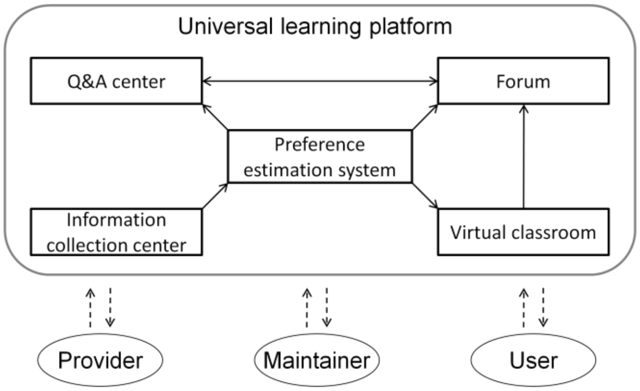

Universal learning platform
Q&A center
Forum
Preference estimation system
Information collection center
Virtual classroom
Provider
Maintainer
User

Fig. 1. The Architecture of the Universal Learning System

II. Overview of the Universal Learning System

There are three kinds of participants in the universal learning system, Provider, Maintainer, and User. These participants share information together, such that the proposed system can work correctly. The role of each kind of participant is described as follows:

1) *Provider:* A provider provides information of hardware kits to the universal learning system, and gets feedback from users. A provider is usually the vendor providing hardware kits, documents, design samples, and consulting. Besides, a provider can be also an educator or learner that provides experience and design samples.

2) *Maintainer:* The maintainer is in charge of maintenance and update of the universal learning system, and transforms information from the provider to helpful statistics for users. For example, the maintainer delivers courses and evaluates hardware kits in the universal learning system. Besides, the maintainer also forward questions or suggestions of users to the provider, so this role can be viewed as a unified window between the provider and users. Therefore, the experts, rich resources, and a good relationship with the provider are three necessary conditions of the maintainer. As an IC research and service center in Taiwan, CIC has sufficient resources and enough experts supported by Taiwan government. Besides, CIC also has a close relationship with both vendors and universities. Thus CIC is very suitable for the role of the maintainer.

3) *User:* A user is able to access information and get suggestions of the universal learning system, take courses, join discussions, ask questions, and share experience and design samples. Besides, it is also important that the provider and maintainer can get feedback from users. Students and teachers are major users of the proposed system.

The purpose of the universal learning system is to help users select a suitable embedded hardware kits according to the requirements, and then provide a friendly learning environment to users. Based on the objective, the proposed system is categorized to five parts, as described below:

1) *The information center collecting dispersed resources together:* While surveying hardware kits, a user usually faces a problem: requested resources (i.e., documents, samples, etc.) are unavailable or dispersed. This is because the resource providers could be different organizations or users, so the user must search for resources through different manners, such as official website, attached CD of the hardware kits, forums, classmates, etc. The universal learning system collects these resources together, so every user can easily get required resources and information on the proposed system.

2) *The preference estimation system for hardware kits:* Another key problem that might be encountered by users is the selection of hardware kits. Since there are numerous hardware kits designed by different vendors for various purposes, the wide diversity usually confuses users. Therefore, the preference estimation system is proposed here to analyze the user preference, compare the features of different hardware kits, and then visualize the relationship between user preference and features of hardware kits. Thus a user can easily understand which hardware kits fulfill his requirements.

3) *The Q&A center for responses of queries:* Frequently asked questions are collected in the question center. Besides, the Q&A service is provided for users. Questions from users will be answered by CIC experts or forwarded to the vendors. Trough the universal learning system, a user can search for the answer of questions and query anytime.

4) *The forum sharing experience and feedback:* The proposed system also provides a forum for users to share their learning experience and report bugs. Vendors can get user responses as their reference to improve their products.

5) *The virtual classroom for online learning:* CIC experts deliver embedded system courses every summer and winter vacation. All video recorded courses can be accessed by users from the universal learning system. Besides, virtual laboratory such as virtual platforms or remote laboratory will be provided in the virtual classroom as well, so that a user can learn both knowledge and practical design experience in the virtual classroom anytime and anywhere. In the future, every user can register on-line as a lecturer or student. A lecturer could give courses and recruit students. CIC will be responsible for the quality of every course.

The architecture of the universal learning system is illustrated in Figure 1. From the selection of hardware kits, learning, laboratory, query, till discussion and sharing experience, the proposed system provides a complete learning environment to users. The arrows shown in Figure 1 represent several possible learning orders of users. For example, a user can have a survey in the information collection center first. Next, the preference estimation center help the user compare the relationship between user preference and hardware features, and then give a suggestion about suitable hardware kits for the user. After the user decides which hardware kit to use, he can take relevant courses in the virtual classroom and join discussion in the forum. The user can also ask questions in the Q&A center if he cannot get required information in the forum.

When a hardware kit and relevant resources are sent to CIC, CIC will start analyzing and evaluating the hardware kit. In the first step, experts of CIC have to understand how to use the hardware kit in depth. Then according to specifications and

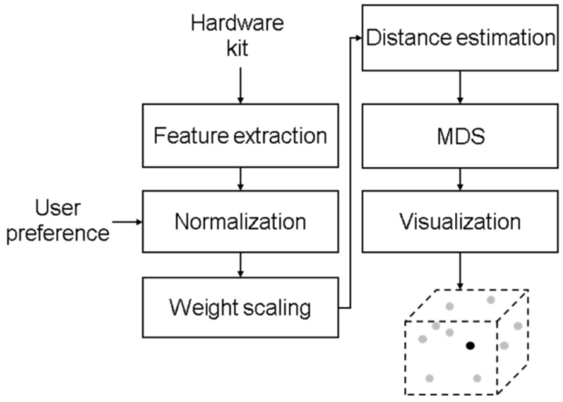


Fig. 2. The Steps To Visualize The Relationship Between Hardware Kits

user experience, the hardware features are extracted into the preference estimation system. After that, design samples will be collected or created by CIC experts in the information collection center. Next, CIC lecturers will deliver related courses. The courses content including lectures and teaching materials will be uploaded to the virtual classroom. The virtual platform is also going to be built up if the vendor is willing to provide detailed specification of the hardware kit to CIC. Meanwhile, an exclusive discussion board will be provided in the forum and question center, such that users can share experience and ask questions. In this article, we put emphasis on description of the preference estimation system because it is the most important part in the universal learning system.

III. Preference Estimation System

The spirit of the universal learning system is the preference estimation system. Before learning embedded systems, the most important thing is selecting a correct hardware kit that corresponds to the user requirements. In this section, we describe the components of the preference estimation system, which has several features:

- Extraction of features from a hardware kit for users.
- Visualization of the relationship between hardware kits.
- Visualization of the user preference.
- Suggestion of suitable hardware kits for users.

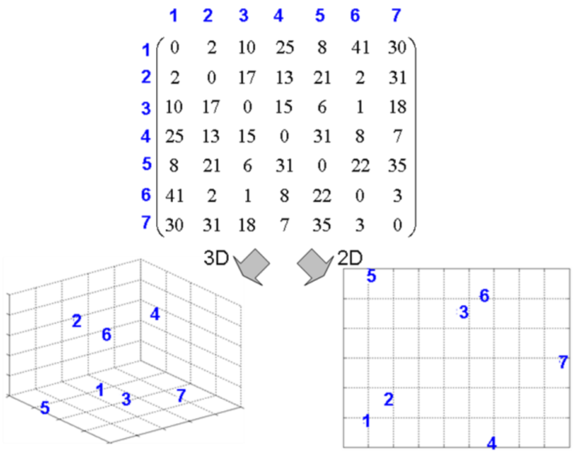
Figure 2 illustrates the flow chart of visualizing the hardware features and user preference. In the proposed system, every hardware kit is mapped to a point in a low dimensional coordinate system, such as a 3D cube, 2D plane, or even a 1D line, where the distance between two points illustrates how similar they are. For this purpose, we have to extract the features of every hardware kit first, and then transform these features to a feature vector. After normalization and weight scaling, these feature vectors are used to measure the distance between each other. Here the Euclidean distance is applied to represent the degree of similarity between two feature vectors. Then we can get a dissimilarity matrix describing the relationship between each pair of feature vectors. However, it is still inconvenient for users to search for relationship in this big matrix, so we use multidimensional scaling [14] to transform the relationship to a low dimensional point, so that users can easily understand this relationship. In the following sub-sections, the details of every step are described.

A. Feature Extraction

In order to select a suitable hardware kit, a user may need to collect the specifications and spend much time comparing different hardware kits. It is a difficult task especially for a beginner. To help users make comparisons and decisions, the proposed system collects useful information and extract features for every hardware kit. These features can be classified as three categories, indicating the general information, complexity, and user experience of a hardware kit, respectively. Several objective and subjective feature factors are included to form a feature vector, listed as follows:

1) General Information: The general information shows the brief specification of a hardware kit. This information let users understand the global view and applicable area of a hardware kit, such as the appearance, size, processor type, peripheral information, etc. Thus the difference between hardware kits becomes clearer and more transparent.

2) Complexity: The complexity includes several factors that could influence the learning performance. Basically we consider five factors in this category, but the factors could be inserted, removed, or updated in practice according to the user feedback. These factors are (1) degree of design transparency, (2) number of design samples, (3) processor core speed, (4) depth of Instruction Set Architecture (ISA), and (5) ease of use of Board-level Support Package (BSP). The first factor “degree of design transparency” means if the design of hardware/software components provided by the vendor is visible. For example, the codec inside a design sample is released to all users without charge. Transparent design benefits deep understanding of the foundational concepts behind design ideas and also helps freeware development. The second factor “number of design samples” means if enough design references can be found. These design references can be provided by vendors, CIC, or common users. More design references means a user can be familiar with the hardware kit more deeply and widely. The third and forth factors are related to the core processor of the hardware kit because many users select hardware kits depending only on the characteristic of a core processor. The factor “core speed” denotes the processor speed, which has significant impact on programming strategies. High-speed cores can be used for complex and general-purposed programming, but low-speed cores are only suitable for simple programming or controlling. On the other hand, high-speed cores usually cost more energy than low-speed ones. In addition to the core speed, “ISA depth” indicating the bit-depth of the instruction set is also important. A long instruction set is usually more complex than a short instruction set. The last factor “ease of use of BSP” means if the board-level support package (BSP) is easy to use. The rating of this factor is evaluated by CIC experts based on their experience. The rated score depends on the user interface, function supports, and integrity of BSP. Note that high complexity does



$$\begin{array}{c|ccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ \hline 1 & 0 & 2 & 10 & 25 & 8 & 41 & 30 \\ 2 & 2 & 0 & 17 & 13 & 21 & 2 & 31 \\ 3 & 10 & 17 & 0 & 15 & 6 & 1 & 18 \\ 4 & 25 & 13 & 15 & 0 & 31 & 8 & 7 \\ 5 & 8 & 21 & 6 & 31 & 0 & 22 & 35 \\ 6 & 41 & 2 & 1 & 8 & 22 & 0 & 3 \\ 7 & 30 & 31 & 18 & 7 & 35 & 3 & 0 \end{array}$$

Fig. 3. An Example Of MDS From 7×7 Dissimilarity Matrix To 3D And 2D Coordinating System

not always imply high difficulty to learn this hardware kits. They are just references for users to select a hardware kit corresponding to the requirements.

3) User Experience: The above two categories are considered from the view of the hardware kit, but it is also important to consider the view of users, so that users can realize how others think about the hardware kit. Therefore, the proposed system allows users voting on-line and leaves comments about every hardware kit.

In fact, the user preference can be expressed by the above factors as well. A user just needs to fill out the requirements of every feature factor. Then the proposed system will estimate the relationship between the user preference and hardware features. Our goal is to visualize this relationship, so that users can easily read the relationship.

In order to construct the relationship between hardware kits, how to represent these feature factors is very important. Since a hardware kit can be described by a feature vector consisting of the feature factors as mentioned above, we quantify these feature factors with numbers. For example, (core number, core speed, number of examples, user rating) can be treated as a feature vector with four feature factors. However, not every feature factor can be quantified. Some factors such as CPU type or peripheral type are only meaningful when they are expressed by text. This kind of feature factors will be excluded during the estimating process.

B. Dissimilarity Matrix

After the extraction of features, we can use a feature vector to describe a hardware kit. With these feature vectors, the relationship between hardware kits can be constructed. Here the relationship can be interpreted as the degree of similarity. Thus, if we let $V_1 = (v_{11}, v_{12}, \ldots, v_{1n})^T$ and $V_2 = (v_{21}, v_{22}, \ldots v_{2n})^T$ be two feature vectors, the simplest way to represent the relationship between V_1 and V_2 is the Euclidean Distance, as shown below:

$$D(V_1,V_2)=\sqrt{(v_{11}-v_{21})^2+(v_{12}-v_{22})^2+...+(v_{1n}-v_{2n})^2} \qquad (1)$$

The sub-index n indicates the number of learning indices. Apparently, a smaller $D(V_1,V_2)$ means a higher degree of similarity between V_1 and V_2. However, the range of every feature factor v_{ij} is different. For example, "user rating" might be scored from 1 to 100, but "the number of design samples" might have only 30 levels. Due to this reason, we have to normalize each feature vector V_i to $V'_i = (v'_{i1}, v'_{i2}, \ldots, v'_{in})$ before the measurement of the Euclidean Distance, such that each V'_i for $i \in [1, n]$ has the same range. Therefore, the above formula becomes:

$$D'(V'_1,V'_2)=\sqrt{w_1(v'_{11}-v'_{21})^2+w_2(v'_{12}-v'_{22})^2+...+w_n(v'_{1n}-v'_{2n})^2} \qquad (2)$$

In (2), w_i is the weighting value indicating the importance of the i-th feature factor. It is configurable when a user prefers one feature factor to others. In general cases, w_i is set to 1 $\forall\, i \in [1, n]$.

Suppose the number of feature vectors is k. After (2) is applied to all pairs of feature vectors, the dissimilarity matrix D can be obtained as follows:

$$D=\begin{pmatrix} D'(V'_1,V'_1) & D'(V'_1,V'_2) & \cdots & D'(V'_1,V'_k) \\ D'(V'_2,V'_1) & \ddots & & \\ \vdots & & & \\ D'(V'_k,V'_1) & & & D'(V'_k,V'_k) \end{pmatrix} \qquad (3)$$

As mentioned before, the dissimilarity matrix D consists of the degree of similarity between each pair of hardware kits. Through the dissimilarity matrix, the relationship among hardware kits becomes obvious.

C. Multidimensional Scaling

In the previous section, we use quantified feature vectors to construct the dissimilarity matrix, which shows the degree of similarity among different hardware kits. Nevertheless, a user still has to spend lots of time searching for the relationship from this $k \times k$ matrix, so a friendly interface to present this relationship is necessary, such that users can read this dissimilarity matrix with ease.

Multidimensional scaling (MDS) is a mathematical tool that uses proximities between objects, subjects, or stimuli to produce a spatial representation of these items. The proximities are defined as any set of numbers that express the amount of similarity or dissimilarity between pairs of objects, subjects or stimuli. It is a data reduction technique because it is concerned with the problem of finding a set of points in low dimension that represents the "configuration" of data in high dimension. Here the "configuration" can be viewed as our dissimilarity matrix D. Then D can be visualized in a low dimensional coordinate system, such as a 2D plane or a 3D cube. Figure 3 shows an example of multidimensional scaling from a dissimilarity matrix to 2D and 3D representation. Suppose there are seven items, noted as item 1 to item 7. In Figure 3, the distance between item i and item j is shown at location (i, j) of the dissimilarity matrix. After multidimensional scaling, the relative position of each item can be visualized in any low-dimensional coordinate systems. Figure 3 illustrates an example of 2D and 3D visualization of a dissimilarity matrix.

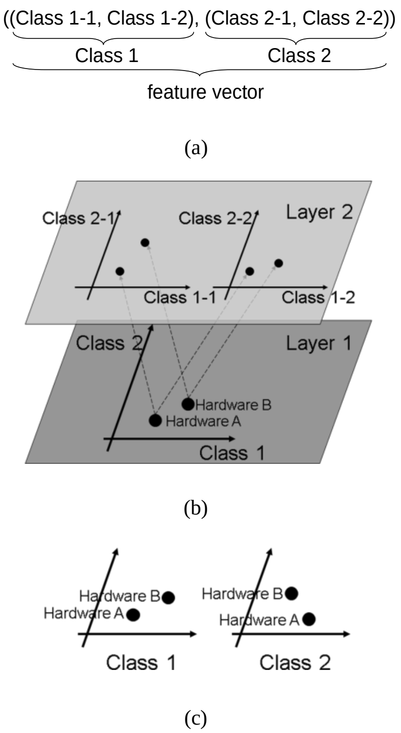


Fig. 4. An example of the preference estimation system

In practice, there are many ways to implement the preference estimation system. For example, the feature factors could be divided to several classes, where each class is visualized by either an entire coordinate system or just an axis. In Figure 4 (a), a feature vector is divided to class 1 consisting of class 1-1 and class 1-2 as well as class 2 consisting of class 2-1 and class 2-2. Figure 4 (b) shows a hierarchical representation to visualize the features of all classes (Layer 1) and subclasses (Layer 2). Here a class or subclass is represented by an axis. Figure 4 (c) is another representation to illustrate class 1 and class 2. In this case, a class is represented by an entire coordinate system. The advantage of this kind of representation is that difference between classes or relationship in a class can be more easily compared. A detailed case study will be described in the next section.

IV. Case Study

As mentioned before, more and more embedded hardware platforms have been developed and used in different courses for various applications. Thus it is unlikely to learn all design skills using one embedded hardware platform. In this case study, three hardware platforms with different features are adopted for embedded system curriculums. The three hardware platforms are imported into the universal learning system, so students can use the proposed system to understand the relationship between the user preference and other feature vectors. For each hardware platform, CIC lecturers deliver a short course, which has been carefully designed, such that the skills required for designing most popular electronic products can be covered in the course. Once a student decides the hardware platform according to the preference, he can take the course either in the CIC classroom or in the virtual classroom.

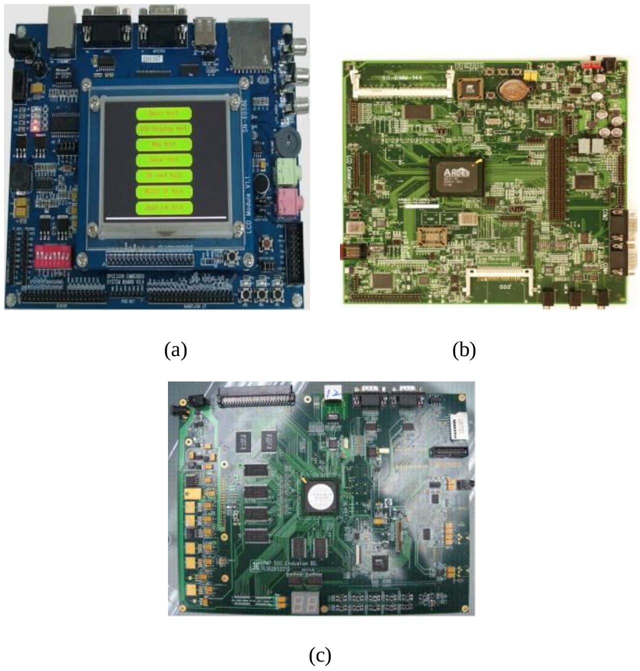


Fig. 5. Three hardware platforms used in the courses: (a) Sunplus SPCE3200, (b) ANDES Leopard, and (c) ITRI PAC-PMP

Besides, he can also join discussion in the forum and ask questions in the Q&A center.

In this section, the three embedded hardware platforms are first introduced. Then the steps to visualize features of the hardware platforms and the user preference are illustrated. Besides, for each hardware platform the course design strategy is described. Finally, student feedback is given for demonstration of the performance of the proposed system.

A. Hardware Platforms

The first hardware platform used in the system is Sunplus SPCE3200 [11], which is a highly integrated platform designed by Sunplus for multimedia applications, as shown in Figure 5 (a). The platform is composed of a domestic 32-bit SoC (S+Core), 128MB SDRAM, 64Mb NOR Flash, 128Mb NAN Flash, QVGA 3.5” TFT LCD, and rich peripherals. The SoC chip of SPCE3200 contains S+core CPU, Advanced High-Performance Bus (AHB) connecting with high-performance modules, and Advanced Peripheral Bus (APB) connecting with low-speed peripheral modules. High-performance modules include CMOS sensor interface (CSI), MPEG-4/JPEG encoding and decoding modules, LCD controller, TV signal encoding module, 2-channel 16-bit D/A converter, embedded 8KB RAM (LDM) and 32KB ROM, and memory interface controller. Low-speed peripheral modules include GPIO controller, SPI serial bus controller, SIO serial bus controller, I2C serial bus controller, I2S master/slave controller, UART controller, USB master/slave controller, Watchdog, SD controller, NAND flash controller, and 9-channel 12-bit A/D converter. A well-integrated development and debugging IDE environment as well as a ported eCOS system is offered with the platform, helpful for the Hardware-Dependent Software (HDS) and application developing. The core of SPCE3200 is S+Core7, a 32-bit and 7-stage pipeline RISC CPU, supporting

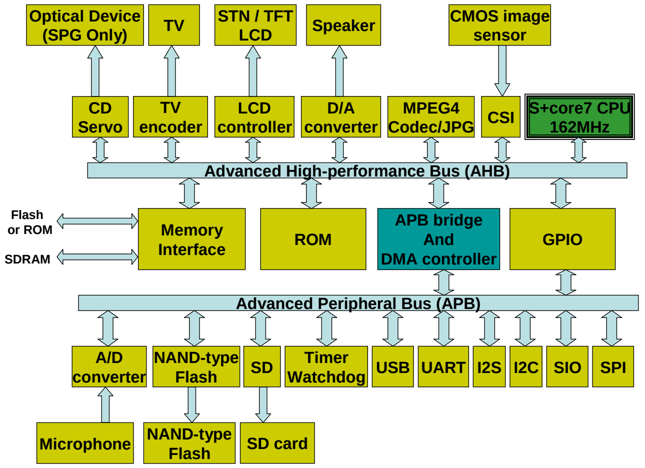


Fig. 6. The Architecture Of Sunplus SPCE3200

16/32-bit mixed instruction set and working at 27~162 MHz. Since many applications can be implemented by hardware, it saves execution time of software programs, and thus achieves low energy cost and high efficiency. Therefore, SPCE3200 is suitable for low-power devices or the controller of devices, such as handheld devices, small household appliances, sensors, or controllers of robots. The architecture of SPCE3200 is shown in Figure 6.

The second hardware platform is ANDES Leopard [12], consisting of a 32-bit SoC, a SODIMM slot of SDRAM, 32MB Flash, QVGA 3.5" TFT LCD, Ethernet, UART, IIC, AC97, SD card, LED, LEDs, buttons, ICE port for on-line debug, and so on, as shown in Figure 5(b). Just like SPCE3200, there are AHB connecting with high-performance modules and APB connecting with low-speed peripheral modules. The core of ANDES Leopard is a 32-bit and 5-stage pipeline SoC, supporting 16/32-bit mixed instruction set. The maximum working frequency of the core can reach 500 MHz. Besides, the chip also contains controller of LCD, Flash, AHB, SDRAM, DMA, and so on. Moreover, Leopard offers a well-integrated development and debugging IDE environment and a well-ported Linux kernel. Since Leopard has a well-ported Linux and a powerful CPU, it is suitable for developing general-purposed applications or applications that requires complex computations, such as Netbooks, mini-computers, or E-books. The architecture of Leopard is shown in Figure 7.

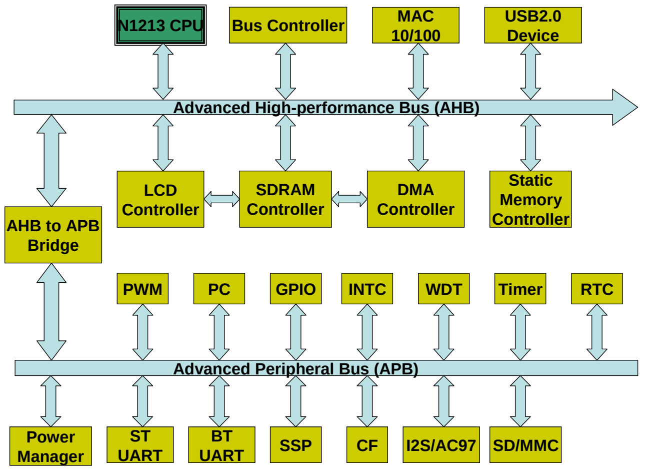


Fig. 7. The Architecture Of ANDES Leopard

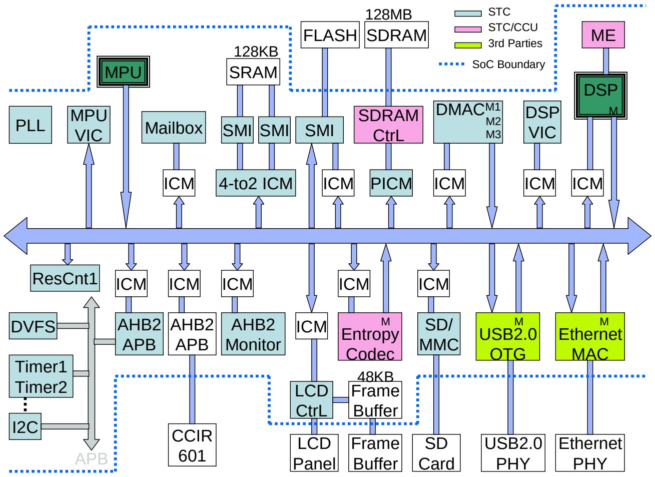


Fig. 8. The Architecture Of ITRI PAC-PMP

The third hardware platform is ITRI PAC-PMP [13] (Figure 5(c)). PAC (Parallel Architecture Core) is a high performance and low energy cost DSP developed by ITRI (Industrial Technology Research Institute of Taiwan). PAC-PMP (Parallel Architecture Core Portable Multimedia Player) is a heterogeneous dual-core hardware platform designed mainly for multimedia and parallel applications. The structure of PAC-PMP is shown in Figure 8. PAC-PMP has a dual-core SoC, 128MB SDRAM, 32MB Flash, VGA TFT LCD, and several interfaces, such as UART, Ethernet, USB, IIC, UART, IrDA, LEDs, IIS, SD card, ICE port for on-line debug, and so on. The core of PAC-PMP adopts the heterogeneous dual-core structure, consisting of ARM926EJ-S and a domestic DSP with 5-way VLIW for multimedia applications. Besides dual-core architecture, the other feature of PAC-PMP is multimedia hardware codec, supporting H.264/AVC hardware accelerator (motion estimation and entropy coding) and multimedia DMA and SDRAM controller. In addition, the SoC also includes the newest Dynamic Voltage Frequency Scaling (DVFS) technique, which can dynamically adjust voltage frequency to efficiently reduce energy cost. Due to the sufficient hardware supports and powerful computational ability, PAC-PMP focuses on high-quality multimedia applications and parallel computing, such as recording video and answering the phone simultaneously. However, the high complexity of its structure also increases the threshold to learn this platform, so PAC-PMP is more suitable for advanced software designer. The applicable area includes PDAs, smart phones, high-definition DVRs, and so on.

B. Preference Estimation

Before taking the courses, a student could provide his requirements (or preference) to the system, and give a weighting value to each feature factor. The requirements of students were treated as a feature vector, just like other hardware platforms. Then we extracted feature factors for each hardware platform. After that, the feature factors were divided into two parts, the ones that can be quantified or not.

TABLE I. UNQUANTIFIED FEATURE FACTORS

	SPCE3200	Leopard	PAC-PMP
Processor	Sunplus S+score	ANDES N1213	PAC DSP + ARM926EJ-S
Memory	128Mb SDRAM, 192Mb Flash	256MB SDRAM SO-DIMM, 32MB NOR Flash	128MB SDRAM, 32MB NOR Flash
Communication Interface	SPI, SIO, IIC, UART, USB, Ethernet, GPRS	Ethernet, UART, IIC	Ethernet, USB, IIC, UART, IrDA
Interaction Interface	Joystick, Touch panel, buttons, LEDs	Touch panel, buttons, LEDs	Touch panel, buttons, LEDs
Multimedia Interface	TV Out, LCD, CMOS camera, IIS	LCD, AC97	LCD, IIS
Other Interface	SD card, SJTAG, GPS	SD card, AHB, X-Bus, MII	SD card, AHB
Applicable area	Peripheral controlling, non-OS programming	OS kernel programming, general-purposed programming	DSP programming, multi-thread programming

Quantified factors were added into the feature vector, and others were listed by a table, as shown in TABLE I. Since the range of every feature factor is different, each factor is normalized to [1, 100] first. Next, every feature vector was adjusted according to the weighting values given by the user, so the result could more approximate the user requirements.

After all feature vectors were extracted and quantified, we categorized the feature vector into three classes, software information, hardware information, and user experience. Each class contains several feature factors. For example, the class of software information contains complexity of BSP, number of design examples, completeness of documents, and so on. The class of hardware information contains the core speed, core size, and some quantified hardware specifications. As for the user experience, it includes user ratings, ease of use, readability of documents, etc. Next, the Euclidean Distance between each pair of feature vectors is calculated to obtain the dissimilarity matrix. After multidimensional scaling of feature vectors in each class, the features of a hardware platform was reduced to three values, standing for software information, hardware information, and user experience, respectively. The three values are expressed by *x*, *y*, and *z* axis of a coordinate system, respectively. Eventually, there will be four points shown in a 3D coordinate system, standing for the feature of each platform and the preference of the user, respectively. Figure 9 shows an example of the preference of a student (U) and the corresponding visualized feature points of the three platforms (S, A, and P).

The distance between two points represents the degree of similarity. Since the distance between ‘S’ and ‘U’ is the shortest, the system suggests the student select SPCE3200. On the other hand, the user could consider from only one dimension (such as the hardware information dimension), so different suggestion might be provided.

C. Course Design

Since every selected hardware platform has different features and applicable areas, we design three short courses to teach these platforms so that students can learn different types of design skills in different courses. Students could choose courses according to the suggestion of the user preference system.

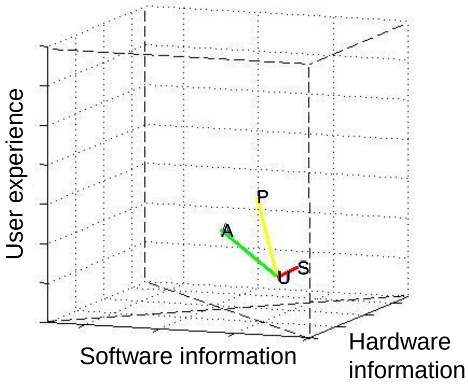


Fig. 9. MDS From Three Hardware Platforms To A 3D Coordinating System, Where S Means Sunplus SPCE3200, A Means ANDES Leopard, P Means ITRI PAC-PMP, And U Means User

The design strategy of courses basically follows ADDIE process [15], which is a systematic framework used for instructional design, and it is also used for helping create new research topics in learning technology. There are five phases in ADDIE process, analysis, design, development, implementation, and evaluation. We simplify these five phases to three steps, instructional objective, instructional strategy, and evaluation, as described in the following subsections.

First, we use ABCD model proposed by Mager [16] to describe our instructional objectives. In ABCD model, useful instructional objectives consist of four necessary components: Audience, Behavior, Conditions, and Degree. According to ABCD model, our instructional objective is defined as: Students (Audience) can learn embedded software design skills (Behavior) and finish 70% lab assignments (Degree) using the simulator and the hardware platform provided in the course (Condition). The details are described as follows:

1) *Audience:* Students are mainly graduated students with basic knowledge of embedded systems, but the community is also accepted to take the course.

2) *Behavior:* Students can learn skills of embedded software design according to their requirements or abilities.

3) *Condition:* Students have to program using the simulator and the hardware platform provided in the class.

4) *Degree:* Students have to finish 70% lab assignments in the course, so that we can make sure how much the students have learned.

The position of the proposed courses is advanced learning of embedded software design, so students are requested to have basic knowledge of embedded systems, operating systems, computer architecture, and C programming. All of the preliminary knowledge can be learned in the university.

Since teaching basic knowledge is omitted, the length of every short course can be curtailed to 15 hours. At beginning of every short course, the basic information of the platform is introduced, including specification of the hardware platform, development tools, peripheral devices and connectors, and

TABLE II. COURSE OVERVIEW

Course Outline		
SPCE3200	***Leopard***	***PAC-PMP***
S+core introduction	ANDES introduction	PAC-PMP SoC overview
IDE tools introduction	Software development using AndesLive	Parallel architecture core - PAC DSP + PAC platform
SPCE3200 platform introduction	AndesStar ISA	Development tools of ARM
SPCE3200 GPIO	ANDES GPIO	Development tools of PAC DSP
SPCE3200 timer	ANDES memory controller	
SPCE3200 interrupt	ANDES timer and interrupt	
	ANDES LCD and DMA	

installation. After the introduction, there are specific topics for different applications in each class. TABLE II illustrates the overview of each course, and TABLE III shows specific topics in each course. For SPCE3200, we focus on design of controller to peripheral programming and low level non-OS device driver programming. For Leopard, the main topic is general purposed software development on Linux. For PAC-PMP, DSP programming and development of dual-core communications and applications are emphasized. Note that the difficulty to learn PAC-PMP is considered as the highest because many students are not familiar with the skills of parallel design and inter-process communication. Therefore, students are suggested to learn parallel programming design before PAC-PMP.

D. Evaluation

Dave addressed five stages composing Psychomotor Domain of Bloom's Taxonomy [17]. The five stages are Imitation, Manipulation, Precision, Articulation, and Naturalization. These stages explained the importance of exercises. Besides, lab exercise is also one of the best ways to evaluate the learning efficiency of a student, so we emphasize on lab exercises in the courses. There are lab exercises assigned after the introduction of every topic. The assignments are designed in order to let students review the topics they just learned. In the first assignment, the teacher shows how to use IDE tools and simple programming skills, and students have to repeat these actions. As for the remaining assignments, the teacher just gives hints, and students have to finish assignments themselves.

In the final lab exercise, students design their own projects according to the given direction. The knowledge they have learned in the courses must be applied in their projects. In the summer courses of 2012, some students integrated their majors into the projects, such as a non-OS MP3 player using SPCE3200, a digital photo frame using ANDES Leopard, and a motion estimation accelerator using PAC-PMP.

TABLE III. SPECIFIC TOPICS OF THE COURSES

Specific Topics		
SPCE3200	***Leopard***	***PAC-PMP***
Programming of peripheral controller	Programming of general purposed applications	DSP programming
Programming of non-OS device driver	Program profiling and optimization techniques	Dual-core project design flow
Programming of tiny-OS applications	Final Lab: Development of applications on Linux	Dual-core project execution flow
Final Lab: Development of applications on eCos or non-OS environments		Final Lab: Usage of AAC, JPEG, and H.264/AVC accelerators

As for the students taking the course in the virtual classroom, several virtual platforms are provided for lab exercise. Most functions on the hardware platform can be simulated by its virtual platform, although the performance might decrease significantly. Another problem is that the student cannot ask question during the lab exercise, but he can join discussion in the forum or query in the Q&A center by email or hotline.

E. Student Feedback

At the end of the courses, we collected 69 questionnaires from students. Before the courses, only 57% students have basic knowledge of embedded systems. There are even 7% students had never touched embedded systems. This means about half of students did not really know their learning objective. After the courses, 98% students agreed the proposed system could more or less help their embedded software design in the future. Among these 98% students, 76% of them felt the proposed system is very helpful. Besides, 88% students believed the courses met their expectation. In ease of use, 100%, 90%, and 65% students agreed that it is easy to use SPCE3200, Leopard, and PAC-PMP, respectively. This result corresponds to our expectation because the preference estimation system reveals that the complex rank of the three hardware platforms is PAC-PMP > Leopard > SPCE3200. A complex hardware platform usually needs more effort to learn. Therefore, from the feedback, it can be concluded the proposed system helped students to select correct hardware platform. In addition, the proposed short courses also helped students to learn embedded software design for different applications.

We also collected the comments from the teachers after the courses. Some comments are quoted as follows:

1) *The visualized interface might attract more students to use the proposed suggesting system. The relation map looks instinctive and easy.*

2) *The proposed system also helps teachers to determine the instructional objectives. The teacher, for example, can map the requirements of all students to coordinate points, and classify students into groups by similarity. For different groups, different assignments or objectives might be given, so that every student could learn according to their requirements and levels. However, the proposed system works only when the student keys in correct user preference. If some students do not really know what they need, the system output will lose accuracy in this situation.*

The main problem from the teachers is inaccurate user input. This error is unavoidable if the user input is inaccurate. Therefore, we suggest the user preference is determined by the feature vector of a platform that the user has used before. For example, in this case study, a user can use the feature vector of SPCE3200 as his preference if he used to use this hardware platform before. Then the system will suggest ANDES to the user because ANDES is more like SPCE3200 than PAC-PMP.

V. Conclusion

In this article, the idea of the universal learning system is proposed. The visualization of the dissimilarity matrix let users intuitively understand the relationship between the hardware platforms and user preference, and helps users select the most suitable hardware platform to learn embedded systems on-line. In addition to online courses and virtual platforms, users can also join discussion and ask questions using the proposed system. In the demonstration, we applied three hardware platforms used for embedded system education in Taiwan. The results show that users can easily select the most suitable hardware platform and take courses according to requirements. Therefore, the proposed system can help users learn embedded system anytime and anywhere.

However, there are still some problems in the proposed system. For example, there should be a well-organized management for the forum. Also, it is difficult to build up the virtual platform for each hardware platform. In the future, these issues will be taken into account gradually, such that the universal learning system can be more practical.

Acknowledgment

The CIC national program was supported in part by National Science Council of Taiwan. The authors wish to thank J.-R. Chang, C.-T. Kuo, and Y.-S. Lin for providing information of Sunplus SPCE3200, ANDES Leopard, and ITRI PAC-PMP.

References

[1] ICT Results (2009). ICT and innovation: From micro-chips to macro-solutions [Online], available at http://cordis.europa.eu/ictresults/index.cfm?section=news&tpl=brochuresreport.

[2] C.-S. Lee, J.-H. Su, K.-E. Lin, J.-H. Chang, and G.-H. Lin, "A project-based laboratory for learning embedded system design With Industry Support," *IEEE Transactions on Education*, vol. 53, no. 2, pp. 173-181, May 2010.

[3] S.H. Kim and J.W. Jeon, "Introduction for Freshmen to Embedded Systems Using LEGO Mindstorms", *IEEE Transactions on Education*, Vol 52, No 1, pp. 99-108.W.-K, Feb. 2009.

[4] M. Winzker and A. Schwandt, "Teaching embedded system concepts for technological literacy," *IEEE International Conference on Microelectronic Systems Education*, pp. 89-92, 2009.

[5] D.T. Rover, R.A. Mercado, Z. Zhang, M.C. Shelley, and D.S. Helvick, "Reflections on Teaching and Learning in an Advanced Undergraduate Course in Embedded Systems", *IEEE Transactions on Education*, Vol 51, No 3, pp. 400-412, Aug. 2008.

[6] J.-S. Chenard, Z. Zilic, and M. Prokic, "A Laboratory Setup and Teaching Methodology for Wireless and Mobile Embedded Systems", *IEEE Transactions on Education*, Vol 51, No 3, pp. 378-384, Aug. 2008.

[7] J. O. Hamblen, "Using a low-cost SoC computer and a commercial RTOS in an embedded systems design course," *IEEE Transactions on Education*, vol. 51, no. 3, pp. 356-363, August 2008.

[8] T. Tierens, P. Pelgrims, W. Dams, and P. V. Pelt, "Interdisciplinary embedded system design in education," *IEEE International Conference on Microelectronic Systems Education*, pp. 135-136, 2007.

[9] S. Nooshabadi and J. Garside, "Modernization of Teaching in Embedded Systems Design – An International Collaborative Project", *IEEE Transactions on Education*, Vol 49, No 2, pp. 254-262, May. 2006.

[10] National Chip Implementation Center, Taiwan. Available at http://www.cic.org.tw

[11] Sunplus Technology Co. Ltd., available at http://www.sunplus.com

[12] ANDES Technology Co. Ltd., available at http://www.andestech.com

[13] Industrial Technology Research Institute, Taiwan, available at http://www.itri.org.tw

[14] I. Borg and P. Groenen, Modern Multidimensional Scaling: theory and applications (2nd ed.), New York: Springer-Verlag, pp. 207–212, 2005.

[15] R. M. Branch, Instructional design: the ADDIE approach, New York: Springer, 2010.

[16] S. Smaldino, D. Lowther, and J. Russell, Instructional media and technologies for learning, 9th ed., Englewood Cliffs: Prentice Hall, Inc, 2007.

[17] R. H. Dave, Developing and writing behavioral objectives, R. J. Armstrong, ed., Tucson, Arizona: Educational Innovators Press, 197.